CSPC2006 DESIGN AND ANALYSIS OF ALGORITHMS (3-0-0)

Course Learning Objectives:

CLO1. Explain the methods of analyzing the algorithms and to analyze performance of algorithms.
CLO 2. State algorithm's efficiencies using asymptotic notations.
CLO3. Solve problems using algorithm design methods such as the brute force method, greedy method, divide and conquer, decrease and conquer, transform and conquer, dynamic programming, backtracking and branch and bound.
CLO4. Choose the appropriate data structure and algorithm design method for a specified application.
CLO 5. Introduce P and NP classes.

Module-I: (8 Hours)

Notion of Algorithm: Growth of functions, Recurrences: The Master method, The Substitution method, The Iteration method, Asymptotic Notations and Basic Efficiency Classes (Use of Big O, θ, etc.) in analysis of algorithms, Mathematical Analysis of few Non-Recursive and Recursive Algorithms.

Module-II: (8 Hours)

Sorting and Searching Techniques: Selection Sort, Bubble Sort, Insertion Sort, Sequential Search, Binary Search, Depth First Search and Breadth First Search, Balanced Search Trees, AVL Trees, Red-Black Trees, Heaps and Heap Sort, Disjoint Set and their Implementation, Divide and Conquer Paradigm of problem solving, Complexity analysis and understanding of Merge Sort, Quick Sort, Binary Search Trees.

Module-III: (8 Hours)

Greedy Techniques: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's and Bellman Ford Algorithm, Huffman Trees, Knapsack problem.
Dynamic Programming Paradigm: Floyd-Warshall Algorithm, Optimal Binary Search trees, Matrix Chain Multiplication Problem, Longest Common Subsequence Problem, 0/1 Knapsack Problem, Maximum
Network Flow Problem.

Module-IV: (8 Hours)

String Matching Algorithms: Naive string-matching algorithm, The Rabin-Karp Algorithm, string matching with Finite Automata, Knuth Morris Pratt string matching algorithm.
Backtracking: n-Queen's problem, Hamiltonian Circuit problem, Subset-Sum problem, State Space Search Tree for these problems

Module-V: (8 Hours)

Branch and Bound: Travelling Salesman Problem and its State Space Search Tree.
Introduction to Computability: Polynomial-time verification, NP-Completeness and Reducibility, NP- Complete problems. Approximation Algorithms: Vertex Cover Problem.

Course outcome:

At the end of the course the student will be able to:
CO1 Analyze the performance of the algorithms, state the efficiency using asymptotic notations and analyze mathematically the complexity of the algorithm.

CO2 Apply divide and conquer approaches and decrease and conquer approaches in solving the problems analyze the same

CO3 Apply the appropriate algorithmic design technique like greedy method, transform and conquer approaches and compare the efficiency of algorithms to solve the given problem.

CO4 Apply and analyze dynamic programming approaches to solve some problems, and improve an algorithm time efficiency by sacrificing space.

CO5 Apply and analyze backtracking, branch and bound methods and to describe P, NP and Complete problems.

Textbooks:

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms", PHI Publication.
2. A.V. Aho, J. E. Hopcroft and J.D. Ullman, "The Design and Analysis of Computer Algorithms", Pearson Education.
3. R. S. Salaria, Khanna, "Data Structure & Algorithms", Khanna Book Publishing Co. (P) Ltd.

Reference Books:

1. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
2. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2nd Edition, 2009. Pearson.

Web links and Video Lectures (e-Resources):

1. https://nptel.ac.in/courses/106/101/106101060/
2. http://cse01-iiith.vlabs.ac.in/